



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/697,867	10/30/2003	David Alan Hepkin	AUS920030356US1	9755

29125 7590 11/15/2005

IBM CORP (JRB)  
C/O LAW OFFICE OF JOSEPH R BURWELL  
P O BOX 28022  
AUSTIN, TX 78755-8022

EXAMINER

IWASHKO, LEV

ART UNIT PAPER NUMBER

2186

DATE MAILED: 11/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/697,867	HEPKIN, DAVID ALAN	
	<b>Examiner</b>	<b>Art Unit</b>	
	Lev I. Iwashko	2186	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 30 October 2003.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

**DETAILED ACTION**

***Claim Rejections - 35 USC § 112***

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 10, 20, and 30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. All three claims stated “the memory page as being in an input/output state after allocating the memory page”. Neither the specification nor the claims defined what it meant to be in an “input/output state”, so it is impossible for someone skilled in the art to enable the invention as stated. Assumptions were made that an “input/output state” meant that the memory page could be accessed.

***Claim Rejections - 35 USC § 102***

2. The following are quotations of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-30 are rejected under U.S.C. 102(b) as being anticipated by Stoodley (US Patent 6,408,305 B1).

Claim 1. A method for initializing a memory page, the method comprising:  
(Abstract, lines 1-4 – State that this is a method for an initializing page)

Art Unit: 2186

- in response to a memory operation by a first thread, allocating a memory page; (*Column 51-55 – State that a page is allocated by the page protection system*)
- generating a request for a second thread to initialize the allocated memory page; (*Column 1, lines 39-48 – Show how the conventional persistent object system “requests” the initialization of the allocated memory page*)
- and initializing the allocated memory page by the second thread in accordance with the request. (*Column 2, lines 2-5 – State that the page is further initialized by swizzling the pointers in the page and allocating new inaccessible pages to swizzled pointers*)

Claim 2. The method of claim 1 further comprising:

- putting the first thread into a sleep state prior to initialization of the allocated memory page; (*Column 2, lines 26-27 – State that threads are stopped while a fault is handled*)
- and in response to completion of initialization of the allocated memory page, putting the first thread into a runnable state. (*Column 2, lines 15-19 – State that the thread can access the page that is accessible after initialization*)

Claim 3. The method of claim 1 further comprising: zeroing the allocated memory page to initialize the allocated memory page. (*Column 3, lines 1-9 – State that root pages are initialized from an object-oriented database where there are access frontier pages. In other words, a first reference follows to a new virtual page, which is the criteria to “zero” the allocated memory page*)

Claim 4. The method of claim 1 further comprising: copying contents from a source page to the allocated memory page to initialize the allocated memory page, wherein the request identifies the source page. (*Column 2, lines 32-40 – State that pages can be transferred from the database into shared*

*memory and the swizzling and initialization is done before making the pages accessible in the program's address space)*

- Claim 5. The method of claim 1 further comprising:
- receiving an interrupt prior to allocating the memory page; *(Column 1, lines 59-65 – State that a page fault (a.k.a. interrupt) is generated when a program attempts to dereference a main memory pointer)*
  - and returning from the interrupt after generating the request for the second thread to initialize the allocated memory page. *(Column 2, lines 2-12 – Show how there is a return from the interrupt after generating the request for the second thread to initialize the allocated memory page)*
- Claim 6. The method of claim 5 further comprising: identifying the interrupt as a result of a page fault. *(Abstract, lines 13-16 – States that there is a page fault handling routine that runs when an interrupt results because of an interrupt)*
- Claim 7. The method of claim 5 further comprising: identifying the interrupt as a result of a copy-on-write fault. *(Column 1, lines 30-35 and lines 55-60 – Reference loading an object using page protection systems of a virtual memory management system. This is the definition given by the inventor of copy-on-write. Also described is the process of paging out main memory pages, which causes an interrupt)*
- Claim 8. The method of claim 1 wherein the memory page is allocated to an application comprising the first thread. *(Column 5, lines 6-9 – State that the main memory maintains a set of objects (a.k.a. thread) of an application program)*
- Claim 9. The method of claim 1 wherein the second thread is a kernel worker thread. *(Column 6, lines 61-65 – State that a deference operation marks the page as accessible subject to the completion of access frontier page processing of the invention. In other words, the off-level worker thread*

*then clears or deletes the page initialization request that it has just completed, and the process is complete)*

Claim 10. The method of claim 1 further comprising:

- indicating the memory page as being in an input/output state after allocating the memory page; *(Column 7, lines 42-53 – State that the page is now an access frontier page after memory allocation)*
- and indicating the allocated memory page as being in a normal state after initializing the memory page. *(Column 6, lines 58-65 – State that the access frontier pages become fully initialized (including complete pointer swizzling) after the initialization process is complete. The page is then marked as accessible (a.k.a. normal state))*

Claim 11. A computer program product on a computer readable medium for use in a data processing system for initializing a memory page, the computer program product comprising: *(Column 1, lines 26-30 – Disclose a computer program)*

- means for allocating a memory page in response to a memory operation by a first thread; *(Column 51-55 – State that a page is allocated by the page protection system)*
- means for generating a request for a second thread to initialize the allocated memory page; *(Column 1, lines 39-48 – Show how the conventional persistent object system “requests” the initialization of the allocated memory page)*
- and means for initializing the allocated memory page by the second thread in accordance with the request. *(Column 2, lines 2-5 – State that the page is further initialized by swizzling the pointers in the page and allocating new inaccessible pages to swizzled pointers)*

Claim 12. The computer program product of claim 11 further comprising:

- means for putting the first thread into a sleep state prior to initialization of the allocated memory page; *(Column 2, lines 26-27 – State that threads are stopped while a fault is handled)*

- and means for putting the first thread into a runnable state in response to completion of initialization of the allocated memory page. (*Column 2, lines 15-19 – State that the thread can access the page that is accessible after initialization*)

- Claim 13. The computer program product of claim 11 further comprising: means for zeroing the allocated memory page to initialize the allocated memory page. (*Column 3, lines 1-9 – State that root pages are initialized from an object-oriented database where there are access frontier pages. In other words, a first reference follows to a new virtual page, which is the criteria to “zero” the allocated memory page*)
- Claim 14. The computer program product of claim 11 further comprising: means for copying contents from a source page to the allocated memory page to initialize the allocated memory page, wherein the request identifies the source page. (*Column 2, lines 32-40 – State that pages can be transferred from the database into shared memory and the swizzling and initialization is done before making the pages accessible in the program’s address space*)
- Claim 15. The computer program product of claim 11 further comprising:
- means for receiving an interrupt prior to allocating the memory page; (*Column 1, lines 59-65 – State that a page fault (a.k.a. interrupt) is generated when a program attempts to dereference a main memory pointer*)
  - and means for returning from the interrupt after generating the request for the second thread to initialize the allocated memory page. (*Column 2, lines 2-12 – Show how there is a return from the interrupt after generating the request for the second thread to initialize the allocated memory page*)
- Claim 16. The computer program product of claim 15 further comprising: means for identifying the interrupt as a result of a page fault. (*Abstract, lines 13-16 –*

*States that there is a page fault handling routine that runs when an interrupt results because of an interrupt)*

- Claim 17. The computer program product of claim 15 further comprising: means for identifying the interrupt as a result of a copy-on-write fault. *(Column 1, lines 30-35 and lines 55-60 – Reference loading an object using page protection systems of a virtual memory management system. This is the definition given by the inventor of copy-on-write. Also described is the process of paging out main memory pages, which causes an interrupt)*
- Claim 18. The computer program product of claim 11 wherein the memory page is allocated to an application comprising the first thread. *(Column 5, lines 6-9 – State that the main memory maintains a set of objects (a.k.a. thread) of an application program)*
- Claim 19. The computer program product of claim 11 wherein the second thread is a kernel worker thread. *(Column 6, lines 61-65 – State that a deference operation marks the page as accessible subject to the completion of access frontier page processing of the invention. In other words, the off-level worker thread then clears or deletes the page initialization request that it has just completed, and the process is complete)*
- Claim 20. The computer program product of claim 11 further comprising:
- means for indicating the memory page as being in an input/output state after allocating the memory page; *(Column 7, lines 42-53 – State that the page is now an access frontier page after memory allocation)*
  - and means for indicating the allocated memory page as being in a normal state after initializing the memory page. *(Column 6, lines 58-65 – State that the access frontier pages become fully initialized (including complete pointer swizzling) after the initialization process is complete. The page is then marked as accessible (a.k.a. normal state))*



- Claim 21. An apparatus for initializing a memory page, the apparatus comprising:  
*(Column 4, lines 23-26 – Declare a computer system that initializes a memory page)*
- means for allocating a memory page in response to a memory operation by a first thread; *(Column 51-55 – State that a page is allocated by the page protection system)*
  - means for generating a request for a second thread to initialize the allocated memory page; *(Column 1, lines 39-48 – Show how the conventional persistent object system “requests” the initialization of the allocated memory page)*
  - and means for initializing the allocated memory page by the second thread in accordance with the request. *(Column 2, lines 2-5 – State that the page is further initialized by swizzling the pointers in the page and allocating new inaccessible pages to swizzled pointers)*
- Claim 22. The apparatus of claim 21 further comprising:
- means for putting the first thread into a sleep state prior to initialization of the allocated memory page; *(Column 2, lines 26-27 – State that threads are stopped while a fault is handled)*
  - and means for putting the first thread into a runnable state in response to completion of initialization of the allocated memory page. *(Column 2, lines 15-19 – State that the thread can access the page that is accessible after initialization)*
- Claim 23. The apparatus of claim 21 further comprising: means for zeroing the allocated memory page to initialize the allocated memory page. *(Column 3, lines 1-9 – State that root pages are initialized from an object-oriented database where there are access frontier pages. In other words, a first reference follows to a new virtual page, which is the criteria to “zero” the allocated memory page)*
- Claim 24. The apparatus of claim 21 further comprising: means for copying contents from a source page to the allocated memory page to initialize the allocated

memory page, wherein the request identifies the source page. (*Column 2, lines 32-40 – State that pages can be transferred from the database into shared memory and the swizzling and initialization is done before making the pages accessible in the program's address space*)

- Claim 25. The apparatus of claim 21 further comprising:
- means for receiving an interrupt prior to allocating the memory page; (*Column 1, lines 59-65 – State that a page fault (a.k.a. interrupt) is generated when a program attempts to dereference a main memory pointer*)
  - and means for returning from the interrupt after generating the request for the second thread to initialize the allocated memory page. (*Column 2, lines 2-12 – Show how there is a return from the interrupt after generating the request for the second thread to initialize the allocated memory page*)
- Claim 26. The apparatus of claim 25 further comprising: means for identifying the interrupt as a result of a page fault. (*Abstract, lines 13-16 – States that there is a page fault handling routine that runs when an interrupt results because of an interrupt*)
- Claim 27. The apparatus of claim 25 further comprising: means for identifying the interrupt as a result of a copy-on-write fault. (*Column 1, lines 30-35 and lines 55-60 – Reference loading an object using page protection systems of a virtual memory management system. This is the definition given by the inventor of copy-on-write. Also described is the process of paging out main memory pages, which causes an interrupt*)
- Claim 28. The apparatus of claim 21 wherein the memory page is allocated to an application comprising the first thread. (*Column 5, lines 6-9 – State that the main memory maintains a set of objects (a.k.a. thread) of an application program*)
- Claim 29. The apparatus of claim 21 wherein the second thread is a kernel worker thread. (*Column 6, lines 61-65 – State that a deference operation marks*

*the page as accessible subject to the completion of access frontier page processing of the invention. In other words, the off-level worker thread then clears or deletes the page initialization request that it has just completed, and the process is complete)*

Claim 30. The apparatus of claim 21 further comprising:

- means for indicating the memory page as being in an input/output state after allocating the memory page; *(Column 7, lines 42-53 – State that the page is now an access frontier page after memory allocation)*
- and means for indicating the allocated memory page as being in a normal state after initializing the memory page. *(Column 6, lines 58-65 – State that the access frontier pages become fully initialized (including complete pointer swizzling) after the initialization process is complete. The page is then marked as accessible (a.k.a. normal state))*

### ***Conclusion***

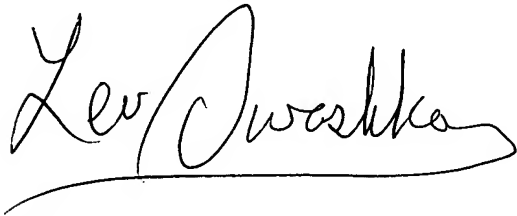
1. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lev I. Iwashko whose telephone number is (571)272-1658. The examiner can normally be reached on M-F (alternating Fridays), from 8-4PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Art Unit: 2186

Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

A handwritten signature in black ink, appearing to read "Lev Iwashko", with a long horizontal flourish extending to the right.

Lev Iwashko

A handwritten signature in black ink, appearing to read "Matthew D. Anderson", with a long horizontal flourish extending to the right.

**MATTHEW D. ANDERSON**  
**PRIMARY EXAMINER**